CSC 261/461
Database Systems

Eustrat Zhupa

November 14, 2018

Indexing

B+ Trees

```
function find(value V)
/* Returns leaf node C and index i such that C.P_i points to first record
* with search key value V */
    Set C = root node
    while (C is not a leaf node) begin
        Let i = smallest number such that V ≤ C.K_i
        if there is no such number i then begin
            Let P_m = last non-null pointer in the node
            Set C = C.P_m
        end
        else if (V = C.K_i)
            then Set C = C.P_{i+1}
        else C = C.P_i /* V < C.K_i */
    end
    /* C is a leaf node */
    Let i be the least value such that K_i = V
    if there is such a value i
        then return (C, i)
        else return null ; /* No record with key value V exists*/
```

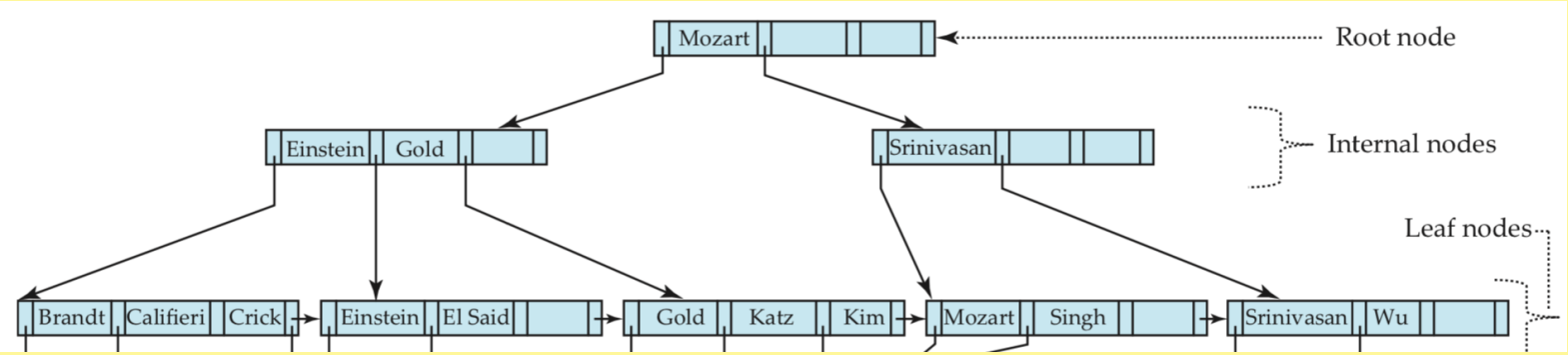
Indexing

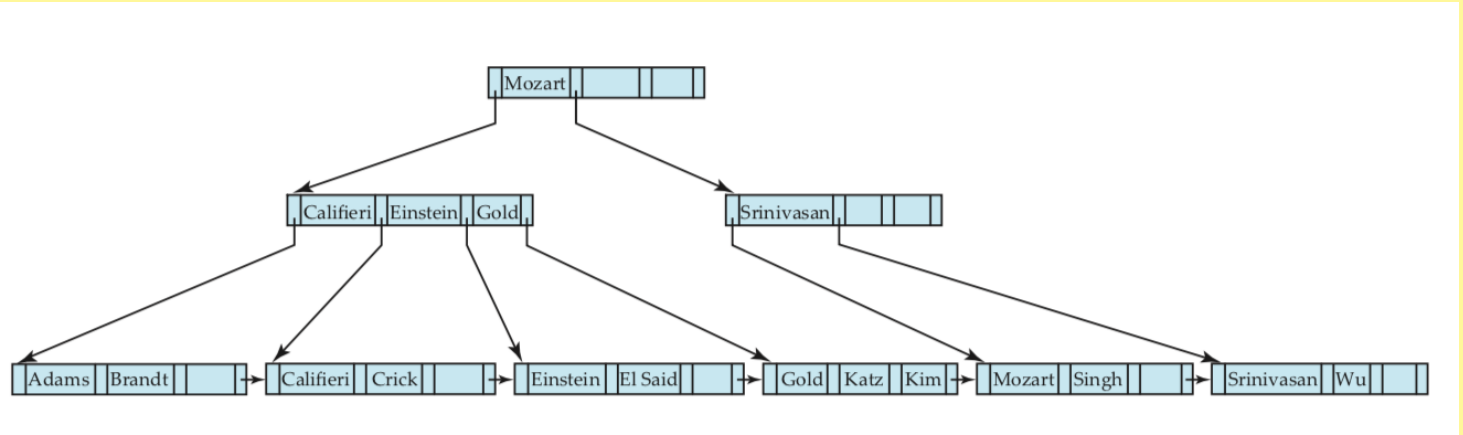
B+ Trees

```
procedure printAll(value V)
/* prints all records with search key value V */
    Set done = false;
    Set (L, i) = find(V);
    if ((L, i) is null) return
    repeat
        repeat
            Print record pointed to by L.P_i
            Set i = i + 1
        until (i > number of keys in L or L.K_i > V)
        if (i > number of keys in L)
            then L = L.P_n
            else Set done = true;
    until (done or L is null)
```

Indexing

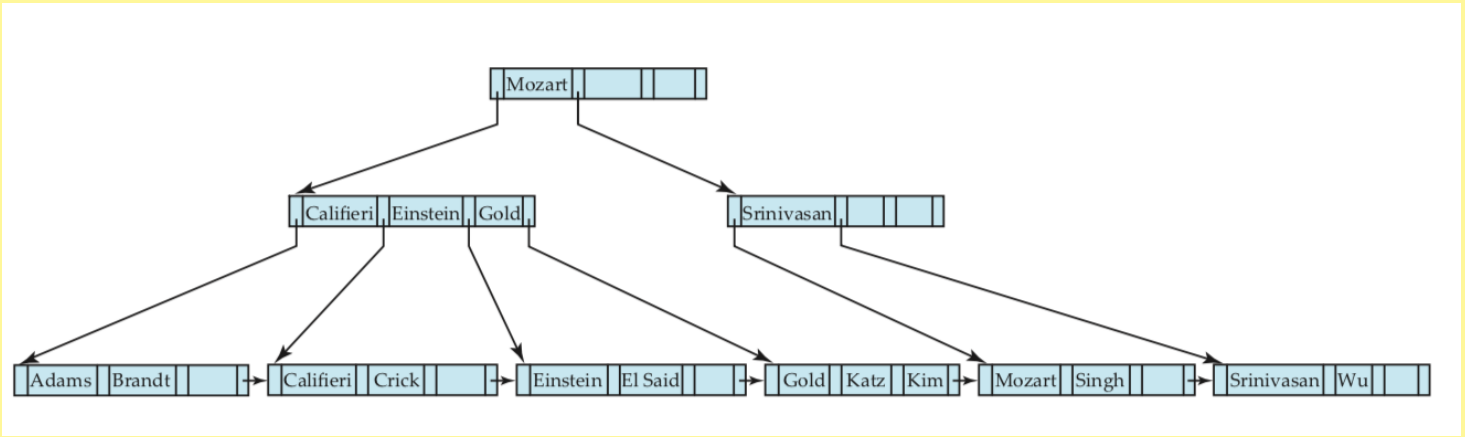
Insertion

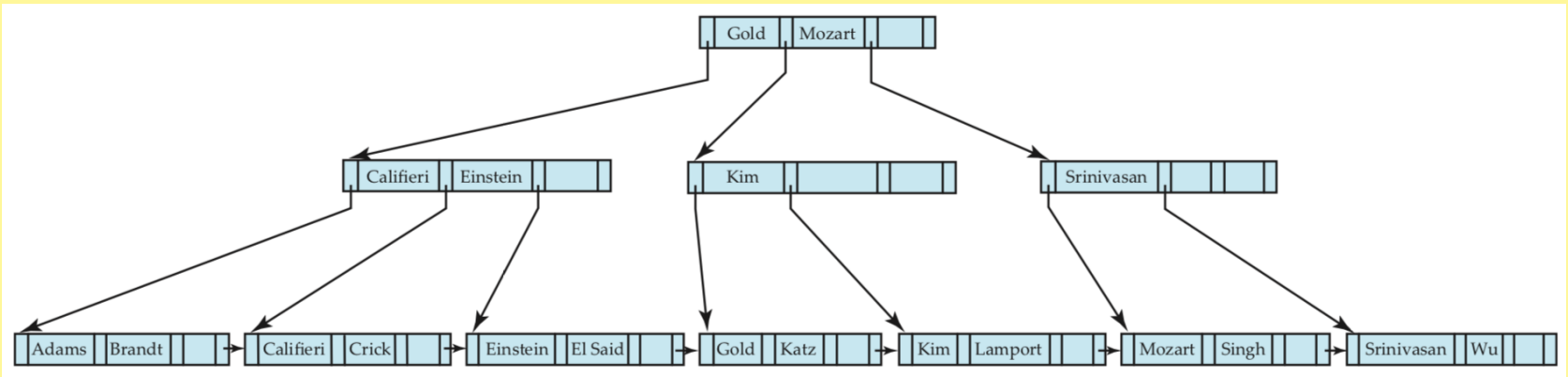




Indexing

Insertion





Hashing Indexes

Hashes

- Disadvantage of sequential file organization:
  - must access an index structure to locate data,
  - must use binary search, and that results in more I/O operations.
- `Hashing` avoids accessing an index structure.
- It provides a way of constructing indices.
  - `bucket` denote a unit of storage that can store records.
  - K is the set of all search-key values
  - B is the set of all bucket addresses.
  - A `hash function` h is a function from K to B.

Hashing Indexes

Hashes

- Hashing can be used for two different purposes.
  1. In a hash file organization, we obtain the address of the disk block through a function.
  2. In a hash index keys are organized into a hash file structure.

Hashing Indexes

Hash Functions

- Worst hash function:
  - maps all search-key values to the same bucket.
  - all the records have to be kept in the same bucket.
  - A lookup has to examine every record to find the one desired.
- Ideal hash function:
  - distributes the stored keys uniformly across all the buckets
  - every bucket has the same number of records.

Hashing Indexes

Hashes

- We want to choose a hash function that assigns search-key values to buckets with a distribution with these qualities:
  - Uniform: That is, the hash function assigns each bucket the same number of search-key values from the set of all possible search-key values.
  - Random: In the average case, each bucket gets same number of values, regardless of the actual distribution of search-key values.

Hashing Indexes

Hashes

bucket 0

bucket 1

15151	Mozart	Music	40000

bucket 2

32343	El Said	History	80000
58583	Califieri	History	60000

bucket 3

22222	Einstein	Physics	95000
33456	Gold	Physics	87000
98345	Kim	Elec. Eng.	80000

bucket 4

12121	Wu	Finance	90000
76543	Singh	Finance	80000

bucket 5

76766	Crick	Biology	72000

bucket 6

10101	Srinivasan	Comp. Sci.	65000
45565	Katz	Comp. Sci.	75000
83821	Brandt	Comp. Sci.	92000

bucket 7

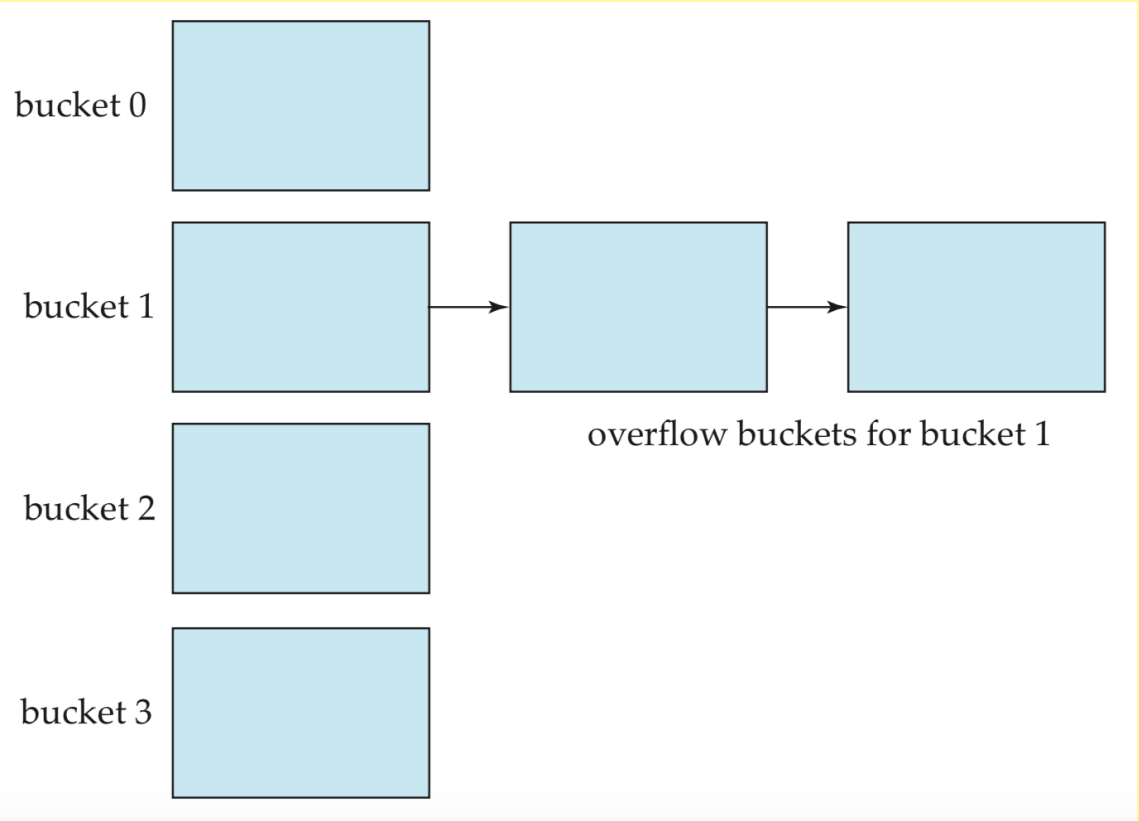
Hashing Indexes

Overflows

- When the bucket does not have enough space, a `bucket overflow`.
- Bucket overflow can occur for several reasons:
  - Insufficient buckets. The number of buckets, must be chosen such that $n_B > n_r/f_r$
  - Skew. Some buckets are assigned more records than are others, so a bucket may overflow even when other buckets still have space.

Hashing Indexes

Hashes



Hashing Indexes

Hashes

- Another form called `open hashing`
  - the set of buckets is fixed,
  - there are no overflow chains. Apply `linear probing`.
  - Drawback: Not flexible and can waste space.

Hashing Indexes

Hash Indexes

- Hashing can be used for index-structure creation.
  - A hash index organizes the search keys, with their associated pointers, into a hash file structure.
    - apply a hash function on a search key to identify a bucket
    - store the key and its associated pointers in the bucket.

Hashing Indexes

Hashes